

**EXAMINER'S AMENDMENT**

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Dan C. Hu (Reg No. 40,025) on 7/2/2010.

Please Amend the Claims As Follows:

1. (Cancelled)
2. (Previously Presented) The method of claim 11, wherein the computer further includes a CPU, wherein the virtual machine monitor is in control of the CPU prior to the runtime virtualization of the I/O device.
3. (Previously Presented) The method of claim 11, wherein the virtualization is performed transparently to an operating system.
4. (Previously Presented) The method of claim 11, wherein the I/O device is compatible with the virtualized I/O device.

5. (Cancelled)
6. (Previously Presented) The method of claim 11, further comprising configuring hardware to trap I/O accesses, and enabling the virtual machine monitor to emulate the I/O device in response to the trapped I/O accesses.
7. (Original) The method of claim 6, wherein the virtual machine monitor uses memory management to trap the I/O accesses.
8. (Previously Presented) The method of claim 15, wherein the virtual machine monitor commences the virtualization between I/O sequences.

9. (Currently Amended) The method of claim 8, ~~wherein the virtual machine monitor commences the virtualization by intercepting I/O accesses;~~ wherein the virtual machine monitor uses the intercepted I/O accesses to update ~~[[a]]~~the state machine, whereby the state machine reflects a state of the I/O device; and wherein the virtual machine monitor examines transitions in the state machine to determine whether the I/O device is in the middle of an I/O sequence.

10. (Previously Presented) The method of claim 15, further comprising the virtual machine monitor commencing the virtualization during an I/O sequence.

11. (Currently Amended) In a computer including an I/O device, a method comprising using a virtual machine monitor to commence virtualization of the I/O device at runtime, wherein runtime is a period of execution in the computer after boot and before shutdown of the computer, wherein virtualization of the I/O device is performed by:

redirecting interrupts for the I/O device to interrupt handlers in the virtual machine monitor;

enabling the virtual machine monitor to intercept I/O accesses by an operating system;

commencing emulation of the I/O device using a state machine;

wherein the virtual machine monitor determines whether the I/O device is performing an I/O sequence, and delays commencing the virtualization until the virtual machine monitor determines that the I/O sequence has completed.

12. (Previously Presented) The method of claim 11, wherein the runtime virtualization includes using the virtual machine monitor to emulate I/O device interrupts.

13. (Currently Amended) The method of claim 11, wherein I/O device interrupts are directed to ~~[[an]]the~~ operating system prior to the runtime virtualization of the I/O device; and wherein the I/O device interrupts are directed to the virtual machine monitor during and after the virtualization of the I/O device.

14. (Currently Amended) The method of claim 11, wherein the virtual machine monitor temporarily pauses ~~[[an]]the~~ I/O sequence by emulating the I/O device as being busy.

15. (Currently Amended) In a computer including an I/O device, a method comprising:

using a virtual machine monitor to commence virtualization of the I/O device at runtime, wherein runtime is a period of execution in the computer after boot and before shutdown of the computer, wherein the I/O device has multiple modes of operations;

the virtual machine monitor determining the mode of the I/O device prior to commencing the virtualization; and

the virtual machine monitor restoring the determined mode of operation after the virtualization, wherein virtualization of the I/O device is performed by:

redirecting interrupts for the I/O device to interrupt handlers in the virtual machine monitor;

enabling the virtual machine monitor to intercept I/O accesses by an operating system; and

commencing emulation of the I/O device using a state machine.

16. (Previously Presented) The method of claim 11, further comprising devirtualizing the I/O device at runtime following the runtime virtualization.

17. (Currently Amended) In a computer including hardware, a method comprising:

running a virtual machine monitor on the hardware;  
running an operating system on the virtual machine monitor,  
wherein the hardware includes an I/O device, and the I/O device is already virtualized by the virtual machine monitor; and  
devirtualizing the I/O device at runtime, wherein runtime is a period of execution in the computer after boot and before shutdown of the computer, wherein devirtualizing the I/O device is performed by:

waiting for a state machine for the I/O device to reach a particular state and blocking new I/O requests;

draining I/O requests queued by the virtual machine monitor; and  
redirecting interrupts for the I/O device to interrupt handlers in the operating system.

18. (Original) The method of claim 17, wherein the devirtualization is performed transparently to the operating system.

19. (Original) The method of claim 17, wherein the devirtualization includes stopping I/O device emulation at runtime.

20. (Original) The method of claim 17, wherein the virtual machine monitor emulates the I/O device prior to devirtualization; and wherein the devirtualization includes allowing the virtual machine monitor to temporarily stop the operating system from commencing a new I/O sequence.

21. (Original) The method of claim 20, wherein the virtual machine monitor temporarily stops the operating system by emulating the I/O device as being in a "busy" or "device not ready" state.

22. (Original) The method of claim 20, wherein the virtual machine monitor bounds the amount of time the operating system processing is temporarily stopped.

23. (Previously Presented) The method of claim 20, further comprising:  
the virtual machine monitor logging I/O accesses by the operating system to the I/O device during devirtualization, and  
replaying the log to the I/O device after devirtualization, wherein the I/O accesses by the operating system are deferred during the devirtualization of the I/O device.

24. (Original) The method of claim 17, wherein the virtual machine monitor waits for I/Os initiated by the virtual machine monitor's driver for the I/O device to complete, and for all expected interrupts from the device to arrive, before ceasing device emulation.

25. (Cancelled)

26. (Previously Presented) The method of claim 17, further comprising, after performing the devirtualizing, configuring the hardware so accesses by the operating system to the I/O device no longer trap to the virtual machine monitor.



27. (Previously Presented) The method of claim 17, wherein the I/O device has multiple modes of operations, the method further comprising:

the virtual machine monitor determining the mode of the I/O device prior to commencing the devirtualization; and

the virtual machine monitor restoring the determined mode of operation after devirtualization.

28. (Previously Presented) The method of claim 17, further comprising virtualizing the I/O device at runtime again after performing the devirtualizing at runtime.

29. (Cancelled)

30. (Previously Presented) The computer of claim 35, wherein the I/O device is compatible with the virtualized I/O device.

31. (Cancelled)

32. (Previously Presented) The computer of claim 35, wherein the hardware is configured to trap I/O accesses, and the virtual machine monitor is enabled to emulate the I/O device in response to the trapped I/O accesses.

Art Unit: 2195

33. (Previously Presented) The computer of claim 32, wherein the virtual machine monitor is configured to use memory management to trap the I/O accesses.

34. (Cancelled)

35. (Currently Amended) A computer comprising:  
hardware including an I/O device; and  
computer memory encoded with a virtual machine monitor for running on the hardware and commencing virtualization of the I/O device at runtime, wherein runtime is a period of execution in the computer after boot and before shutdown of the computer, wherein the virtualization of the I/O device is performed by:

redirecting interrupts for the I/O device to interrupt handlers in the virtual machine monitor;

enabling the virtual machine monitor to intercept I/O accesses by an operating system;

commencing emulation of the I/O device using a state machine,

wherein the virtual machine monitor is configured to determine whether the I/O device is performing an I/O sequence, and to delay commencing the virtualization until the virtual machine monitor determines that the I/O sequence has completed.

36. (Previously Presented) The computer of claim 35, wherein the virtual machine monitor is configured to temporarily pause the I/O sequence by emulating the I/O device as being busy.

37. (Previously Presented) The computer of claim 35, wherein the runtime virtualization includes using the virtual machine monitor to emulate I/O device interrupts.

38. (Currently Amended) A computer comprising:  
hardware including an I/O device; and  
computer memory encoded with a virtual machine monitor for devirtualizing the I/O device at runtime, wherein runtime is a period of execution in the computer after boot and before shutdown of the computer, wherein devirtualization of the I/O device is performed by:  
waiting for a state machine for the I/O device to reach a particular state  
and blocking new I/O requests;  
draining I/O requests queued by the virtual machine monitor; and  
redirecting interrupts for the I/O device to interrupt handlers in the  
operating system.

39. (Previously Presented) The computer of claim 38, wherein the virtual machine monitor is configured to emulate the I/O device prior to commencing the devirtualization; and wherein the virtual machine is configured to commence the devirtualization by temporarily stopping an operating system running on the virtual machine monitor from commencing a new I/O sequence.

40. (Previously Presented) The computer of claim 39, wherein the virtual machine monitor is configured to temporarily stop the operating system by emulating the I/O device as being in a "busy" or "device not ready" state.

41. (Previously Presented) The computer of claim 39, wherein the virtual machine monitor is configured to bound the amount of time the operating system processing is temporarily stopped.

42. (Previously Presented) The computer of claim 38, wherein the virtual machine monitor is configured to log I/O accesses by an operating system to the I/O device during devirtualization, and to replay the log to the I/O device after devirtualization.

43. (Previously Presented) The computer of claim 39, wherein the virtual machine monitor is configured to wait for I/Os initiated by a virtual machine monitor driver for the I/O device to complete, and for all expected interrupts from the I/O device to arrive, before ceasing device emulation.

44. (Previously Presented) The computer of claim 38, wherein the hardware is configured so operating system accesses to the I/O device no longer trap to the virtual machine monitor after the devirtualization.

45. (Previously Presented) The computer of claim 38, wherein the I/O device has multiple modes of operations; wherein the virtual machine monitor is configured to determine the mode of the I/O device prior to commencing the devirtualization; and wherein the virtual machine monitor is configured to restore the determined mode of operation after the I/O device has been devirtualized.

46. (Previously Presented) The computer of claim 38, wherein the virtual machine monitor is configured to further virtualize the I/O device after having devirtualized the I/O device at runtime.

47. (Cancelled)

48. (Previously Presented) The article of claim 52, wherein the virtualization includes commencing I/O device emulation at runtime.

49. (Previously Presented) The article of claim 48, wherein the virtual machine monitor configures the hardware to trap I/O accesses, and enables the virtual machine monitor to emulate the I/O device in response to the trapped I/O devices.

50. (Previously Presented) The article of claim 49, wherein the virtual machine monitor uses memory management to trap the I/O accesses.

51. (Cancelled)

52. (Currently Amended) An article for a computer including an I/O device, the article comprising computer-readable memory encoded with a virtual machine monitor for causing the computer to commence virtualization of the I/O device at runtime, wherein runtime is a period of execution in the computer after boot and before shutdown of the computer, wherein virtualization of the I/O device is performed by:

redirecting interrupts for the I/O device to interrupt handlers in the virtual machine monitor;

enabling the virtual machine monitor to intercept I/O accesses by an operating system;

commencing emulation of the I/O device using a state machine.

wherein the virtual machine monitor determines whether the I/O device is performing an I/O sequence, the virtual machine monitor delaying the commencement of the virtualization until the virtual machine monitor determines that the I/O sequence has completed.

53. (Original) The article of claim 52, wherein the virtual machine monitor temporarily pauses the I/O sequence by emulating the I/O device as being busy.

54. (Previously Presented) The article of claim 52, wherein the virtual machine monitor emulates I/O device interrupts during the runtime virtualization.

55. (Cancelled)



56. (Currently Amended) An article for a computer including an I/O device, the article comprising computer-readable memory encoded with a virtual machine monitor for causing the computer to devirtualize the I/O device at runtime, wherein runtime is a period of execution in the computer after boot and before shutdown of the computer, wherein devirtualizing the I/O device is performed by:

waiting for a state machine for the I/O device to reach a particular state  
and blocking new I/O requests;

draining I/O requests queued by the virtual machine monitor; and  
redirecting interrupts for the I/O device to interrupt handlers in the  
operating system.

57. (Original) The article of claim 56, wherein the devirtualization includes ceasing emulation of the I/O device at runtime.

58. (Previously Presented) The article of claim 57, wherein the devirtualization includes temporarily stopping an operating system running on the virtual machine monitor from commencing a new I/O sequence.

59. (Original) The article of claim 58, wherein the virtual machine monitor temporarily stops the operating system by emulating the I/O device as being in a "busy" or "device not ready" state.

60. (Original) The article of claim 58, wherein the virtual machine monitor bounds the amount of time the operating system processing is temporarily stopped.

61. (Previously Presented) The article of claim 57, wherein the virtual machine monitor waits for I/Os initiated by a virtual machine monitor driver for the I/O device to complete, and for all expected interrupts from the I/O device to arrive, before ceasing device emulation.

62. (Previously Presented) The article of claim 56, wherein the virtual machine monitor logs I/O accesses by an operating system to the I/O device during devirtualization, and replays the log to the I/O device after devirtualization.

63. (Previously Presented) The article of claim 56, wherein the virtual machine monitor, configures the hardware so operating system accesses to the I/O device do not trap to the virtual machine monitor.

64. (Previously Presented) The article of claim 56, wherein the I/O device has multiple modes of operations; and wherein the virtual machine monitor determines the mode of the I/O device prior to commencing devirtualization; and restore the determined mode of operation after the I/O device has been devirtualized.

65. (Cancelled)

66. (Previously Presented) The article of claim 56, wherein the virtual machine monitor causes the computer to further virtualize the I/O device after having devirtualized the I/O device at runtime.

-- END OF AMENDMENT --

### ***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ERIC C. WAI whose telephone number is (571)270-1012. The examiner can normally be reached on Mon-Fri, 9am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng - Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eric C Wai/  
Examiner, Art Unit 2195

/Li B. Zhen/  
Primary Examiner, Art Unit 2194